# Audio Phylogenetic Analysis Using Geometric Transforms

Sebastiano Verde, Simone Milani
Dept. of Information Engineering
University of Padova, Padova, Italy

Paolo Bestagini, Stefano Tubaro
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Milan, Italy

***Abstract*—Whenever a multimedia content is shared on the Internet, a mutation process is being operated by multiple users that download, alter and repost a modified version of the original data leading to the diffusion of multiple near-duplicate copies. This effect is also experienced by audio data (e.g., in audio sharing platforms) and requires the design of accurate phylogenetic analysis strategies that permit uncovering the processing history of each copy and identify the original one. This paper proposes a new phylogenetic reconstruction strategy that converts the analyzed audio tracks into spectrogram images and compare them using alignment strategies borrowed from computer vision. With respect to strategies currently-available in literature, the proposed solution proves to be more accurate, does not require any a-priori knowledge about the operated transformations, and requires a significantly-lower amount of computational time.**

## I. Introduction

Creating and sharing multimedia objects has become easier than ever in the last few years. Any smartphone in everyone's pocket enables shooting images, recording videos and audio tracks, as well as editing any multimedia content already available. Through the use of social networks, multimedia sharing platforms, or personal web pages, it is then possible to share any kind of content at worldwide level. On one hand, this process allows fast and broad information sharing, which is paramount in many situations (e.g., newscasts, reporting of terroristic attacks, etc.). On the other hand, in order to avoid undesired diffusion of illicit and maliciously forged material, this process has highlighted the need of techniques for multimedia content forensic analysis [1]. For this reason, many methodologies to verify authenticity and integrity of images [2], [3], videos [4] and audio tracks [5] have been proposed in the literature.

Many of the proposed forensic solutions work by analyzing each multimedia object as a single entity. These techniques are fundamental when no additional contextual information is available on the content under analysis. However, in the multimedia sharing scenario, when content related to some event of interest are to be analyzed (e.g., political speeches, accidents documented by multiple users, etc.), it is possible to perform additional forensic evaluations by leveraging the availability of multiple near-duplicate (ND) objects, i.e., edited versions of the same original material. This joint analysis of ND multimedia content is known as multimedia phylogeny [6].

Multimedia phylogeny has been initially proposed for image analysis [6]–[8]. In this context, a set of techniques have been proposed to reconstruct the so called phylogeny tree (PT), i.e., an oriented loop-free graph depicting parent-child relationship among all ND images under analysis. This tree basically represents which image has been used to generate the other ND copies through editing operations. Being able to reconstruct the PT enables to study the way content has spread. Moreover, the root of the tree represents the original object that generated all the others. Therefore, PT knowledge also permits restricting additional forensic analysis on the original content, rather than on copies.

In the last few years, despite the advance of image phylogeny [9]–[11], additional work has been carried out on video phylogeny as well [12]–[14]. However, little effort has been put toward audio phylogenetic approaches. As a matter of fact, to the best of our knowledge, the only algorithm proposed in the literature is represented by [15].

In this paper, we focus on audio phylogeny, proposing an algorithm to estimate audio phylogeny tree (APT) from the analysis of a pool of ND audio tracks. In particular, the proposed approach works by considering audio excerpts as images in the time-frequency domain. In doing so, it is possible to map audio editing operations into geometric transforms [16] for efficient audio tracks comparison. With respect to the state-of-the-art algorithm in [15], the proposed approach has two main benefits: i) it can easily deal with temporal and pitch audio transformations not considered in [15]; ii) it is more efficient avoiding the time consuming brute force approach of [15].

The rest of the paper is structured as it follows. Section II presents the problem of phylogenetic analysis for digital audio tracks. Section III describes the adopted strategy for dissimilarity computation. Section IV overviews the whole

This work has been partially supported by the University of Padova project Phylo4n6 prot. BIRD165882/16. This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.

*WIFS'2017, December, 4-7, 2012, Rennes, France.*
*978-1-5090-6769-5/17/$31.00* 

algorithm, whose performances are discussed in Section V. Section VI draws the final conclusions.

## II. Audio Phylogenetic Analysis: Problem Statement

Let us consider an original audio track represented by a monodimensional signal $x_0(n)$. All audio excerpts obtained from this original track by applying one or more editing operations (e.g., pitch shifting, time stretching, fading, trimming, compression, to mention some of the most commonly used) are considered near-duplicates (NDs) [15]. The goal of audio phylogenetic analysis (and this paper) is to process an unordered set $\mathcal{A}$ of $N$ near-duplicate digital audio tracks $x_i(n),\ i = 0, ..., N-1$, to estimate their processing history and mutual relations. This is done by reconstructing a tree structure $\mathcal{T}$, called *Audio Phylogeny Tree* (APT), whose paths going from the root to the different nodes describe the relative generative sequence.

The rationale behind APT reconstruction is that, given two ND tracks $x_i(n)$ and $x_j(n)$, if the former has been generated from the latter through non invertible operations, two conditions must hold: i) it is possible to write $x_j(n) = f_{\boldsymbol{\beta}}[x_i(n)]$, where $\boldsymbol{\beta}$ denotes the set of control parameter values of the audio editing transform $f$; ii) it is not possible to write the vice versa, i.e., obtain $x_i(n)$ from $x_j(n)$. Given a set of $N$ tracks, it is then possible to estimate the associated APT $\hat{\mathcal{T}}$ by verifying these conditions for all audio pairs according to a *dissimilarity* function $d_{i,j}$ defined as

$$d_{i,j} = \min_{\boldsymbol{\beta}} \mathcal{L}\left[x_j(n),\ f_{\boldsymbol{\beta}}\left[x_i(n)\right]\right], \tag{1}$$

where $\mathcal{L}$ is any distance metric. The so-defined dissimilarity is a measure of how likely it is possible to obtain $x_j(n)$ from $x_i(n)$. Once dissimilarity has been computed for all $N \times (N-1)$ audio track pairs, a *dissimilarity matrix* $D = [d_{i,j}]$ is built, where $d_{i,j}$ is referred to the pair of audio tracks $x_i(n)$ and $x_j(n)$. Dissimilarity matrix represents a graph where nodes are audio tracks that are linked through dissimilarity values. From the analysis of $D$ through tree reconstruction algorithms [17], it is then possible to estimate $\hat{\mathcal{T}}$.

The most computationally-intensive and crucial step in the APT reconstruction is dissimilarity computation. As a matter of fact, neither the editing operation $f$, nor the parameters $\boldsymbol{\beta}$ are known in advance by the analyst. Moreover, small errors in dissimilarity computation may propagate in huge mistakes at tree reconstruction levels (e.g., parent-child inversion).

To the best of our knowledge, the only APT reconstruction algorithm proposed in the literature [15] adopts an exhaustive brute-force computation strategy that tests different transform functions $f$ to estimate $d_{i,j}$. Unfortunately, this solution proves to be prohibitive in terms of computational complexity and presumes that the set of adopted transformations is completely known. In the following section, we show how it is possible to overcome these problems and estimate $d_{i,j}$ avoiding brute-force approaches.

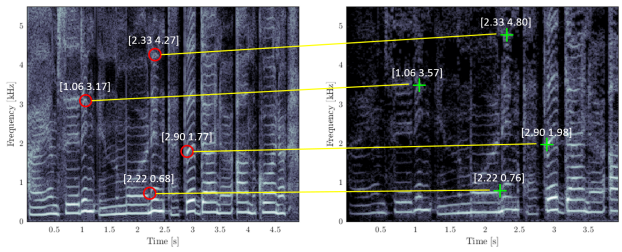


Fig. 1. Spectrograms $P_i(u,v)$ (left) and $P_j(u',v')$ (right) where $x_j(n)$ was generated applying a pitch shift, a fading (at the beginning), and a trimming (at the end) to $x_i(n)$. The two graphs report some of the the matched SURF keypoints $\left((u_k, v_k), (u'_h, v'_h)\right)$.

## III. Estimating Dissimilarity via Spectrogram Analysis

In this paper, we propose a new analysis approach based on computer-vision registration strategies that permit overcoming computational issues presented by [15]. The core idea consists in representing audio tracks by converting the associated spectrograms into bi-dimensional images; the alignment of the spectrograms of $x_i(n)$ and $x_j(n)$ permits estimating an affine geometric transformation modeled by the $3 \times 3$ matrix $H$, whose values can be directly related to the adopted $f_{\boldsymbol{\beta}}(\cdot)$ [16]. As a consequence, it is possible to partially compensate $f_{\boldsymbol{\beta}}(\cdot)$ and obtain a more accurate value of $d_{i,j}$.

The following sections will describe this process in more detail.

### A. Time-frequency representation of audio tracks

At first, each audio track $x_i(n)$ is converted into a 2D signal by computing the short-time Fourier transform

$$X_i(f,m) = \sum_{n=-\infty}^{+\infty} x_i(n)\ w(n-mL)\ e^{-j2\pi fn} \tag{2}$$

where $w(\cdot)$ is a windowing function and L is the stride parameter. Coefficients $X_i(f,m)$ are computed for a finite set of $N_f$ frequencies $f$ ($f = 0, \ldots, F_c - F_c/N_f$) and a finite set of windows ($m = 0, \ldots, M-1$). Associating each coefficient to the pixel of a grayscale image $P_i(u,v)$[1], we obtain a $N_f \times M$ graylevel image, where the pixel intensity is obtained by converting the value $|X_i(f,m)|^2$ into an 8-bit integer (see Fig. 1). In order to remove part of the background noise, if $|X_i(f,m)|^2 < \delta$, the pixel $P_i(u,v)$ is set to 0.

At this point, it is easy to notice that most of the 1-D transformations $f_{\boldsymbol{\beta}}$ employed in the generation of ND tracks ($x_j(n) = f_{\boldsymbol{\beta}}[x_i(n)]$) bijectively correspond to geometric transformations $F_{\boldsymbol{\beta}}$ on $P_i(u,v)$, i.e., $P_j(u',v') = F_{\boldsymbol{\beta}}[P_i(u,v)]$. Fig. 1 reports a simple example: a pitch shift on $x_i(t)$ results in a vertical stretching of the spectrogram (where frequencies are distributed on a linear scale) and an initial fading results in a progressive dimming of pixel intensities going leftward.

[1]Note that $u = f/F_c$ and $v = m$.

This duality suggests compensating $f_{\boldsymbol{\beta}}$ in the domain of spectrograms $P_i(u,v)$ estimating $F_{\boldsymbol{\beta}}$: in this way, it is possible to exploit the availability of highly-optimized computer vision libraries and obtain a more accurate estimate of $f_{\boldsymbol{\beta}}$ (since multiple transformations can be estimated at the same time).

### B. Geometric alignment

Some of the modifications can be modeled by an affine transformation on the domain of $P_i(u,v)$, i.e.,

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = H \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_T & 0 & t \\ 0 & s_P & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (3)$$

where pixel locations $(u,v)$ are expressed in homogeneous coordinates. The affine transformation matrix $H$ presents 3 non-trivial values since time stretching is controlled by parameter $s_T$, pitch shift by parameter $s_P$, and $t$ models the temporal misalignment. As a matter of fact, estimating $H$ permits understanding whether one of these transformations were applied (if none was used, $s_T = s_P = 1$ and $t = 0$).

This estimation can be performed by computing a set of keypoints $\mathcal{K}_i = \{(u_k, v_k)\}$ and their relative descriptors on every spectrogram $P_i(u,v)$; in our implementation, we adopted SURF descriptors [18]. By matching the descriptors of $\mathcal{K}_i$ and $\mathcal{K}_j$, it is possible to associate the pixel $(u_k, v_k)$ of $P_i$ to the pixel $(u'_h, v'_h)$ of $P_j$. As a result, it is possible to have a list of matched pairs $((u_k, v_k), (u'_h, v'_h))$, which can be used in eq. (3) to find $H$. In order to remove outliers and wrong matches, the estimation is performed using the RANSAC algorithm.

After estimating $H$, it is possible to align $P_i$ on $P_j$ generating $P_i^r$ and use the MSE between $P_j$ and $P_i^r$ as a dissimilarity measurement. Note that $H$ estimation accuracy depends on the amount of keypoints found on $P_i$ and $P_j$; this number is affected by the threshold $\delta$, and therefore, its value is optimized on the dataset $\mathcal{A}$ in order to maximize the minimum number of detected keypoints, i.e.,

$$\delta = \arg\max_{\delta} \min_{x_i \in \mathcal{A}} |\mathcal{K}_i|$$

It is also worth noticing that spreading the frequencies $f$ on a logarithmic scale would have mapped a pitch shift operation into a simple frequency translation (i.e., coefficient in position $(2,3)$ of $H$ would be non-null rather than $s_P$) as explained in [16]. However, most of the keypoints are usually located at low frequencies where logarithmic and linear scales are similar. Moreover, in the logarithmic scale, keypoints result distributed on a smaller image areas leading to less robust estimation of the affine transformation $H$. This fact has been confirmed by experimental results, thus linear scale is finally adopted and pitch shift is estimated as linear stretching of low-frequencies components.

### C. Intensity equalization

Also the intensity values of $P_i$ are affected by other modifications on $x_i(n)$. Fig. 1 shows that fading leads to a progressive dimming of pixel intensity going towards image boundaries. Similarly, MP3 coding produces a strong attenuation and the appearance of artifacts at high frequencies, i.e., on the upper part of image $P_i$.

These are extremely useful when the estimated geometric transform leads to an ambiguity in the causal relation between $P_i$ and $P_j$, i.e., the alignment of $P_i$ on $P_j$ leads to the same dissimilarity of the reverse alignment. Therefore, in case geometric alignment can not clearly reveal whether $P_i$ is more likely to be the parent of $P_j$ or vice versa, the spectrogram with less dimmed intensities or artifacts is considered as an ancestor of the other one.

In the following section, we will describe how these equalizations were included in the overall phylogenetic analysis.

## IV. The proposed strategy

Every phylogenetic analysis algorithm can be divided into two main steps: the computation of dissimilarities and the estimation of the phylogenetic tree $\hat{\mathcal{T}}$.

### A. Dissimilarity computation

The relations between pairs of audio tracks can be well modeled by a dissimilarity matrix $D = [d_{i,j}]$, where $d_{i,j}$ models the divergence between audio excerpts $x_i(n)$ and $x_j(n)$. In our approach, the parameter $d_{i,j}$ can be computed as follows:

1) Generate the spectrogram images $P_i(u,v)$ and $P_j(u',v')$, compute SURF descriptors $\mathcal{K}_i$, $\mathcal{K}_j$, and match them.
2) Given a few matched pairs $((u_k, v_k), (u'_h, v'_h))$, compute the homography matrix $H$ exploiting eq. (3) with RANSAC.
3) Extract the values $s_T$ and $s_P$ from $H$, operate the transformation on $x_i(n)$ (in the time domain) creating the signal $x_i^c(n)$.
4) Generate the spectrogram image $P_i^c(u,v)$ from $x_i^c(n)$, compute SURF descriptors $\mathcal{K}_i^c$, and match them to $\mathcal{K}_j$ again.
5) Compute the new homography $H^c$ and transform the spectrogram image $P_i^c$ in $P_i^r$ using $H^c$.
6) Estimate the final dissimilarity

$$d_{i,j} = \frac{1}{U \cdot V} \sum_{u,v} \|P_j(u,v) - P_i^r(u,v)\|^2, \quad (4)$$

where $U$ and $V$ are spectrograms height and width in pixels.

Note that temporal and pitch transformations are corrected in point 3 in the temporal domain for two main reasons: i) the used spectrogram image does not include phase information, thus working directly in time domain allows better pitch-shift and time-stretch control; ii) correct pitch-shifting directly in the spectrogram domain means working with log-frequencies, or applying a log-stretch, which might be a trickier solution leading to less stable results. Additionally, note that since $s_T$ and $s_P$ have already been compensated at point 3, matrix $H^c$ is likely to have values close to 1 along the diagonal. The term $t$ in $H^c$ is instead non-trivial ($\neq 0$), and can be used to align

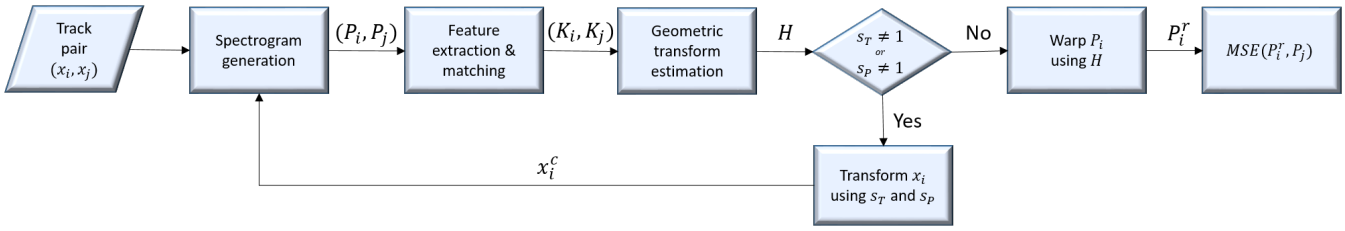


Fig. 2. Block diagram of spectrogram alignment procedure.

and crop the two spectrograms, thus compensating any time shift and trimming operation.

In case $d_{i,j} = d_{j,i}$, a further check is employed. Applying the same considerations reported in Section III-C, the algorithm verifies whether the intensity or the high-frequency artifacts are lower on $P_j$. If so, it is more probable that $x_j$ is a descendant of $x_i$, and therefore, the algorithm set $d_{j,i} = +\infty$.

The whole strategy is reported in the block diagram of Fig. 2.

The matrix $D$ is then processed by a Minimum Spanning Arborescence (MSA) estimation strategy in order to find out the underlying APT $\hat{\mathcal{T}}$, as it will be described in the following subsection.

### B. Audio phylogenetic tree estimation

Starting from the dissimilarity matrix $D$, it is possible to build a complete directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $N$ nodes, where the $i$-th node is associated to the audio track $x_i(n)$. Each edge $(i, j)$ is then labeled with weight $d_{i,j}$, and the spanning arborescence $\hat{\mathcal{G}} = \{\mathcal{V}, \hat{\mathcal{E}}\}$ with minimum weight is then computed finding the subset $\hat{\mathcal{E}} \subset \mathcal{E}$ s.t. from root node/track $r$ to all the others a unique path exists and

$$\hat{\mathcal{E}} = \arg \min_{\mathcal{E}^s \subset \mathcal{E}} \sum_{(i,j) \in \mathcal{E}^s} d_{i,j}$$

is minimum. The root node $r$ corresponds to a $j$-th node s.t. $\nexists (i, j) \in \hat{\mathcal{E}}$.

In our implementation, $\hat{\mathcal{G}}$ is found via the Chu-Liu/Edmonds' optimum branching algorithm [19], [20] and it can be associated to an audio phylogeny tree $\hat{\mathcal{T}}$. Fig. 3 reports an example on a small set of 4 audio tracks.

The following section will evaluate the effectiveness of the proposed approach.

## V. Experimental results

In this section we report all details about dataset creation and algorithm validation, also considering comparison against the state-of-the-art [15].

### A. Dataset

The proposed algorithm has been tested on a dataset built generating multiple near-duplicate trees from tracks of different genres and length. Specifically, we used the following five audio tracks as tree roots:

1) Excerpt from "Ludwig Thuille: *Piano Sextet in B-flat major, Op. 6 - III*"[2] (26 s).

[2]https://www.jamendo.com/track/146588/

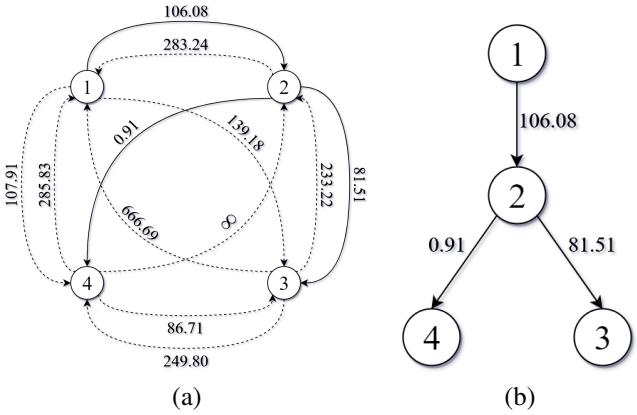


Fig. 3. Estimation of APT $\hat{\mathcal{T}}$ using the optimum branching algorithm on a 4 track set. (a) Initial complete graph $\mathcal{G}$; (b) the estimated arborescence $\mathcal{G}^*$

2) Electric guitar blues riff [3] (34 s).
3) Hand-crank music box playing *Amazing Grace*[4] (31 s).
4) Male voice reading verses from Edgar Allan Poe's *The Raven*[5] (41 s).
5) MIDI loop from *Super Mario*[6] (15 s).

The set of considered audio transformation to build ND tracks consists of:

1) Trim, applied to the leading or trailing samples, with a maximum length of 3 seconds;
2) Fade, applied to the leading (fade-in) or trailing (fade-out) samples, maximum length of 3 seconds;
3) Time stretching, speed-up or slow-down, up to 10%;
4) Pitch shifting, one semitone up or down;
5) MP3 coding, implemented with the LAME encoder, quality factor $q = \{2, 3, 4\}$.

All transformations and relative parameters listed above were picked randomly in the ND generation process in order to test the algorithm on generic phylogenetic trees.

From these premises, we constructed different phylogeny trees according to two strategies:

- *single transformation dataset* - for each track, we generated 10 trees of 50 nodes each, with a single audio transformation per edge.

[3]https://www.freesound.org/s/30021/
[4]https://www.freesound.org/s/180384/
[5]https://www.freesound.org/s/189467/
[6]https://www.freesound.org/s/179684/

- *multiple transformations dataset* - for each track we generated additional 10 trees of 50 nodes each with multiple transformations per edge (up to 4).

Our dataset thus consists of 100 phylogeny trees for a total amount of 5,000 audio excerpts.

### B. Algorithm Validation

The first performed experiment consisted in reconstructing dataset trees, varying the total number of nodes, $K$, from 10 to 50. This was implemented by randomly pruning a set of $50-K$ nodes from the whole trees in the dataset, starting from the leaves and moving upwards. We conducted the experiment separately on the single and multiple transformations datasets, in order to find out whether the algorithm encountered difficulties on one case with respect to the other. Reconstructed trees have been evaluated by comparison with the related ground-truth trees according to the *Root*, *Edges*, *Leaves* and *Ancestry* metrics [6]. Denoting with $\mathcal{T}$ and $\hat{\mathcal{T}}$ the ground-truth and estimated APT, respectively, the metrics are defined as follows:

$$Root(\mathcal{T},\hat{\mathcal{T}}) = \begin{cases} 1, & \text{if } root(\mathcal{T}) = root(\hat{\mathcal{T}}) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

$$Edges(\mathcal{T},\hat{\mathcal{T}}) = \frac{|\mathcal{E} \cap \hat{\mathcal{E}}|}{N-1}, \tag{6}$$

$$Leaves(\mathcal{T},\hat{\mathcal{T}}) = \frac{|L \cap \hat{L}|}{|L \cup \hat{L}|}, \tag{7}$$

$$Ancestry(\mathcal{T},\hat{\mathcal{T}}) = \frac{|A \cap \hat{A}|}{|A \cup \hat{A}|}, \tag{8}$$

where $N$ is the number of nodes, $\mathcal{E}$ represents tree edges, $L$ tree leaves and $A$ nodes ancestral relationships. In other words, *Root* metric is 1 if the root is correctly determined and 0 otherwise; *Edges* represents the percentage of correctly estimated edges; *Leaves* represents the percentage of correctly identified leaves; *Ancestry* evaluates the percentage of correctly identified ancestors for each node of the tree.

In Fig. 4 and Fig. 5 we show the results obtained for the single and multiple transformations datasets, respectively. As we can see, except for the smallest tree size ($K = 10$) in which the single-transformation dataset seems to provide slightly better results, the two scenarios are comparable. In particular, the *Root* metric is stable at 98% in the first case, and a bit more noisy in the second one, while remaining at around 96% on average. These results show that the algorithm is not particularly strained by the presence of a higher number of processing operations. Also, such results are quantitatively compatible to those obtained in the framework of image phylogeny [6].

As an additional experiment, we tested our algorithm on trees where certain nodes were randomly removed (always preserving the root). In this way, we can evaluate its performances in a more realistic scenario, that is where only a subset of the whole phylogeny tree is available to the analyst. Performances are expected to decrease as we increase the number of removed

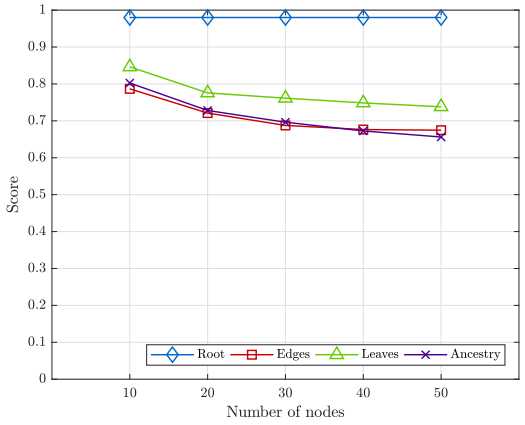


Fig. 4. R-E-L-A metrics for different tree sizes, single transformation per edge.

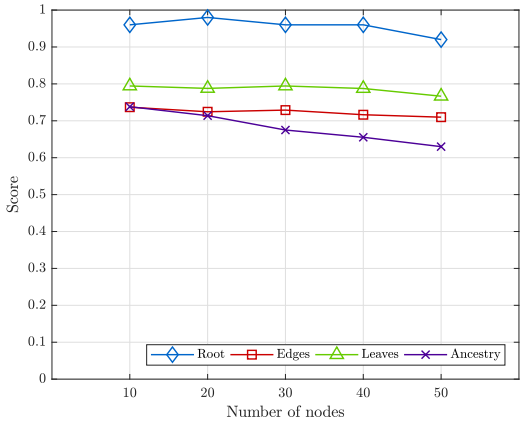


Fig. 5. R-E-L-A metrics for different tree sizes, multiple transformations per edge.

nodes. We conducted this test with up to 25 removed nodes (half the size of the whole tree). Results are reported in Fig. 6. As expected, we observe an overall decrease of about 10% for the *Edges*, *Leaves* and *Ancestry* metrics. *Root* metric instead does not present a decreasing trend, remaining approximately constant at around 95%.

### C. Comparison Against State-of-the-Art

In addition to assessing accuracy results of our method under different conditions, we also performed a comparison against the baseline method presented in [15]. Specifically, we run both algorithms on 40 trees of 10 nodes each, obtained using the single transformation strategy for dataset generation.

The algorithm in [15] performs a brute-force search on a candidate set of audio transforms and parameters. As candidate transforms we selected all editing operations actually used to generate the dataset. As parameters for the brute-force grid search, we selected three candidates for compression (i.e., the ones used for dataset generation), 20 for fading (i.e., uniformly sampling 0 to 3 seconds at either track's head or tail), 20 for

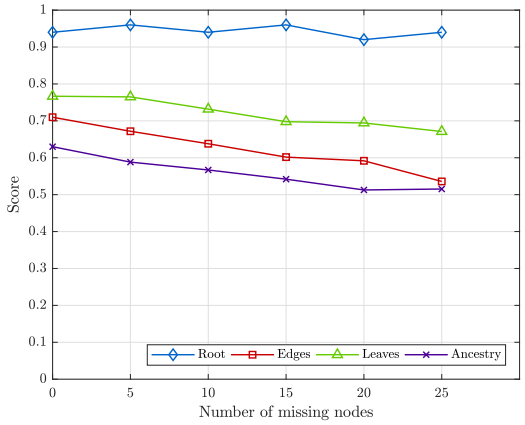


Fig. 6. R-E-L-A metrics for increasing numbers of removed nodes (root always preserved).

TABLE I
COMPARISON AGAINST BASELINE SOLUTION [15]. THE PROPOSED SOLUTION ACHIEVES HIGHER ACCURACY WITH AVERAGE PROCESSING TIME FOR A 10 NODES TREE THAT IS LESS THAN A HALF.

| | Root | Edges | Leaves | Ancestry | Time |
|---|---|---|---|---|---|
| baseline [15] | **97.5%** | 71.7% | 78.3% | 77.5% | 436 s |
| proposed | **97.5%** | **76.1%** | **81.4%** | **79.3%** | **203 s** |

time stretching (uniformly sampling the range used for dataset generation) and 8 for pitch shifting (from -4 to +4 semitones). We implemented the fastest version of [15], which only search for a single transformation from node to node coherently with the used dataset.

Results are reported in Tab. I. It is possible to notice that the proposed approach always performs slightly better (or on par) with the baseline. Moreover, we also considered the average processing time needed to process a tree with Matlab implementations of both algorithms run on a MacBook Pro equipped with a 2.2 GHz Intel Core i7, 8 GB of RAM and SSD disk. This test confirms that our solution is more efficient being able to process each tree in less than half the time needed by [15].

## VI. CONCLUSIONS

In this paper we proposed a solution to the audio phylogeny tree (APT) reconstruction problem. Given a set of near-duplicate (ND) audio tracks, our algorithm is able to reconstruct causal relationships among all audio segments, enabling a precise APT estimation. Differently from state-of-the-art techniques, our approach do not involve brute-force search of possible audio editing operations applied to audio tracks. Moreover, we do not need to know in advance a specific set of candidate audio transformations. Conversely, we leverage computer-vision techniques for image registration applied to audio tracks in the time-frequency domain, which can be in principle applied to any one-dimensional signal affected by time-frequency transformations.

A validation campaign performed on a wide set of trees built starting from audio excerpts of different genres, confirms that the proposed solution enables faster APT reconstruction, still with very accurate performance compared to [15].

Future work will be focused toward the study of spectrogram effects due to different audio processing operations, such as equalization, compression and so on, in order to make the testing environments closer to real ND audio sets.

## REFERENCES

[1] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.
[2] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Computing Surveys (CSUR)*, vol. 43, pp. 26:1–26:42, 2011.
[3] A. Piva, "An Overview on Image Forensics," *ISRN Signal Processing*, vol. 2013, pp. 1–22, 2013.
[4] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, "An overview on video forensics," *APSIPA Transactions on Signal and Information Processing*, vol. 1, p. e2, 2012.
[5] R. C. Maher, *Overview of Audio Forensics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 127–144.
[6] Z. Dias, A. Rocha, and S. Goldenstein, "First steps toward image phylogeny," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2010.
[7] L. Kennedy and S.-F. Chang, "Internet image archaeology," in *ACM international conference on Multimedia (ACMMM)*, 2008.
[8] A. De Rosa, F. Uccheddu, A. Costanzo, A. Piva, and M. Barni, "Exploring image dependencies: A new challenge in image forensics," in *SPIE Conference on Media Forensics and Security*, 2010.
[9] N. L. Philippe, W. Puech, and C. Fiorio, "Phylogeny of JPEG images by ancestor estimation using missing markers on image pairs," in *International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2016.
[10] M. Oikawa, Z. Dias, A. Rocha, and S. Goldenstein, "Manifold Learning and Spectral Clustering for Image Phylogeny Forests," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 11, pp. 5–18, 2016.
[11] S. Milani, P. Bestagini, and S. Tubaro, "Phylogenetic analysis of near-duplicate and semantically-similar images using viewpoint localization," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016.
[12] J. R. Kender, M. L. Hill, A. Natsev, J. R. Smith, and L. Xie, "Video genetics," in *ACM Conference on Multimedia*, 2010.
[13] Z. Dias, A. Rocha, and S. Goldenstein, "Video Phylogeny: Recovering near-duplicate video relationships," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2011.
[14] F. Costa, S. Lameri, P. Bestagini, Z. Dias, S. Tubaro, and A. Rocha, "Hash-based frame selection for video phylogeny," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016.
[15] M. Nucci, M. Tagliasacchi, and S. Tubaro, "A phylogenetic analysis of near-duplicate audio tracks," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2013.
[16] X. Zhang, B. Zhu, L. Li, W. Li, X. Li, W. Wang, P. Lu, and W. Zhang, "Sift-based local spectrogram image descriptor: a novel feature for robust music identification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, p. 6, 2015.
[17] Z. Dias, S. Goldenstein, and A. Rocha, "Exploring heuristic and optimum branching algorithms for image phylogeny," *Journal of Visual Communication and Image Representation*, vol. 24, pp. 1124–1134, 2013.
[18] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008, similarity Matching in Computer Vision and Multimedia.
[19] J. Edmonds, "Optimum branchings," *J. Res. Nat. Bur. Standards*, vol. 71B, pp. 233–240, 1967.
[20] Y. J. Chu and T. H. Liu, "On the shortest arborescence of a directed graph," *Science Sinica*, vol. 14, p. 1396?1400, 1965.